Raspodijeljene glavne knjige i kriptovalute

Bitcoin protokol

Ante Đerek, Zvonko Konstanjčar

29. listopada 2021.

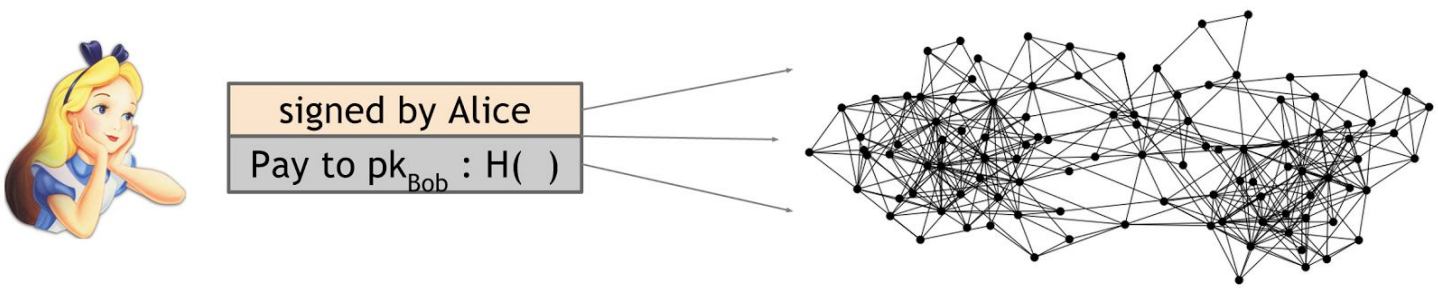
Prva vježba: Bitcoin skripte

- Ako radite u paru, prijava do 31.10.2021. u 23:59h.
- Rok za predaju putem MS Teams-a: 07.11.2021. u 23:59h.
- Konzultacije: putem MS Teams-a po dogovoru (`rgkk@fer.hr`)
- Ocjenjivanje: izvorni kod, obrana uživo

Fakultet elektrotehnike i računarstva

Arhitektura sustava

- Puno čvorova u "peer-to-peer" mreži.
- Svi čvorovi imaju skoro identične kopije lanca blokova.
- Periodički se dodaje novi blok u lanac:
  - Čvor koji riješi kriptografsku slagalicu predloži sljedeći blok.
  - Svaki čvor provjeri ispravnost bloka prije prihvaćanja.



Izvor: `bitcoinbook.cs.princeton.edu`

Proof-of-work sustav

- Blok je ispravan ako sadrži rješenje kriptografske slagalice.
- Čvorovi preferiraju najdulje ispravne blokove.
- Čvorovi se nagrađuju za predlaganje bloka fiksnom nagradom i naknadama za transakcije.

Ako je hash funkcija H korisna za slagalice onda je najbolji mogući algoritam rudarenja ...

Za zadane transakcije $x_1, x_2, \ldots, x_n$ i *prag* t

1. Izaberi novi broj *nonce*.
2. Izračunaj $h = H(nonce|hash_{prev}|x_1|x_2|\ldots|x_n)$.
3. Ako je $h < t$ predloži blok, inače idi na početak.

Zadatak

Neka je fiksiran prag t i neka čvorovi $A_1, A_2, \ldots, A_n$ rudare tako da čvor A_i računa v_i sažetaka po sekundi.

- *Koliko je očekivano vrijeme da čvor A_i riješi slagalicu?*
- *Koliko je očekivano vrijeme da neki čvor predložiti novi blok?*
- *Koja je vjerojatnost da će čvor A_i prvi riješiti slagalicu i predložiti sljedeći blok?*

Iz perspektive pojedinog rudara x:

- t_{next} je očekivano vrijeme prije nego što x predloži blok.
- u_x je omjer računalnih resursa rudara x i svih rudara.

$$t_{next} \approx \frac{10 \text{ minuta}}{u_x}$$

Bitcoin (listopad 2021.)

- Reward Per Block – 6.25+0.08112 BTC (405,265.68 USD)
- Difficulty – 20,082,460,130,830
- Hashrate – 150.685 Ehash/s
- Bitcoin Mining Profitability – 0.4707 USD/Day for 1 THash/

Bitcoin (listopad 2018.)

- Reward Per Block – 12.50+0.1439 BTC (83,521.13 USD)
- Difficulty – 7,454,968,648,263

Zadatak

Može li napadač ukrasti ili potrošiti tuđi novčić?

Zadatak

Može li napadač uskratiti uslugu određenom korisniku?

Zadatak

Može li napadač vlastiti novčić potrošiti dvaput?

Zadatak

Može li napadač narušiti povjerenje u sustav?

Pretpostavimo da napadač kontrolira većinu računalnih resursa (51% napad).

Poticaji i "proof-of-work" su riješili sve naše probleme!

Dobili smo mehanizam biranja slučajnog bloka.

- Nemoguće je predvidjeti kojem će se bloku posrećiti da riješi slagalicu i predloži sljedeći blok.
- Ako ispravni čvorovi kontroliraju većinu računalnih resursa onda je vjerojatnost da je čvor koji predlaže sljedeći blok ispravan veća od pola.

Svojstva Nakamotovog konsenzusa

- Kriptografija štiti od krađe i neispravnih transakcija.
- Konsenzus štiti od dvostrukog trošenja.
- Konsenzus je implicitan.
- Konsenzus je vjerojatnosni.

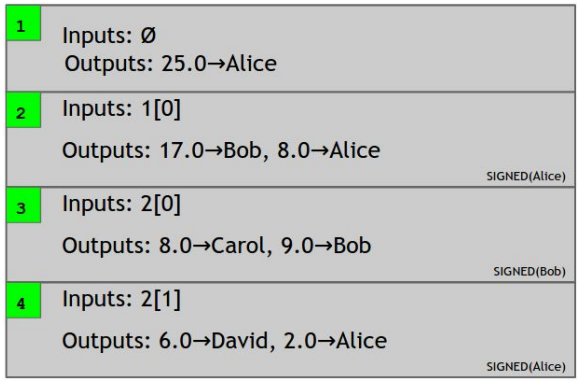
Kako iznutra izgleda sustav Bitcoin?

- Bitcoin transakcije / skripte / blokovi / mreža.

Specifikacija?

- `en.bitcoin.it/wiki/Protocol_documentation`
  This page describes the behavior of the reference client. The Bitcoin protocol is specified by the behavior of the reference client, not by this page.
- Izvorni kod *reference* klijenta: `github.com/bitcoin/bitcoin/`
- *Bitcoin Improvement Proposals*: `github.com/bitcoin/bips/`

Svaka transakcija troši postojeće i stvara nove novčiće:



Izvor: `bitcoinbook.cs.princeton.edu`

Moguće je lagano ostvariti:

- Provjeru je li novčić već potrošen.
- "Djelomično" trošenje novčića (*change address*).
- "Spajanje" novčića.
- Zajednička plaćanja.

Unutar lanca blokova i prilikom komunikacije na Bitcoin mreži:

- Transakcija je zapisana u kompaktnom binarnom obliku.
- Hash transakcije služi kao njezin identifikator (dvaput SHA256, *little endian*).

```
0200000002ded7cf00846ce3003bf1abd870bf0278563c4372272f3d043c760f9d0811e6e
9010000006b483045022100ff591639dc151585efb5a06a18ee33358a8c08b3c5ce38de63
06d3670aa7c4c002201f1491776b4926839c6efdce6d39e0485479fb193bf401c450f4bdd
25d9541ba01210209c6fd112e7588303cd9022d280786885b6f5a8e8ca4e8c1746dbcb249
0d8ad8feffffffdb1f82098d0f74c38bba730d308f506e0ecdf2f24fe93c9986191d48b42
41e4e000000006a47304402205372c5f9d465e3852c7b7781a41e095c5886feffd75bad77
4fd4740334bb296f02203fce4a696f27d6889af035d02bdcf1272224b099faaac8e0e83bf
5146c616164012103d1c0598252e7ab49377e45dbfbf133ed4676258c0958be363e4e4ad8
539dda97feffffff024b200e00000000001976a9140d8a36dc80076d1d8d4ffd4681a9dfb
4265b85e288ac42981700000000001976a9146c86f34ad14d62eec62a4f8a5140a5418504
413888ac83550800
```

Bitcoin transakcije – *metadata*


```
"txid": "19d9abb1f1658472ff6ef366ae05d6b450b727e70ef0639ef43b3ed9f7f3
"hash": "19d9abb1f1658472ff6ef366ae05d6b450b727e70ef0639ef43b3ed9f7f3
"version": 2,
"size": 373,
"vsize": 373,
"locktime": 546179,
"vin": [
    ...
],
"vout": [
    ...
]
```

```
"vin": [
  {
    "txid": "e9e611089d0f763c043d2f2772433c567802bf70d8abf13b00e36c84
    "vout": 1,
    "scriptSig": { ... },
    "sequence": 4294967294
  },
  {
    "txid": "4e1e24b4481d1986993ce94ff2f2cd0e6e508f300d73ba8bc3740f8d
    "vout": 0,
    "scriptSig": { ... },
    "sequence": 4294967294
  }
]
```

Bitcoin transakcije – *outputs*


```
"vout": [
  { "value": 0.00925771,
    "n": 0,
    "scriptPubKey": { ... },
  },
  { "value": 0.01546306,
    "n": 1,
    "scriptPubKey": { ... },
  }
]
```

Do sada smo smatrali da transakcije sadrže:

- Transakcija t_1 – *output* x – `pubKey`: Javni ključ vlasnika.
- Transakcija t_2 – *input* y – `sig`: Potpis transakcije.

Pravilo trošenja:

Novčić (t_1, x) se može potrošiti kao *input* (t_2, y) ako *input* (t_2, y) u polju `sig` sadrži potpis bitnih dijelova transakcije t_2, koji se uspješno provjerava javnim ključem u polju `pubKey` novčića (t_1, x):

$$V(\texttt{pubKey}, t_2', \texttt{sig}) = \texttt{true}.$$

Poopćenje:

- Transakcija t_1 – *output* x – `scriptPubKey`: *locking* skripta.
- Transakcija t_2 – *input* y – `scriptSig`: *unlocking* skripta.

Pravilo trošenja:

Novčić (t_1, x) se može potrošiti kao *input* (t_2, y) ako se spajanjem skripte `scriptSig` *input*-a (t_2, y) i skripte `scriptPubKey` *output*-a (t_1, x) dobiva skripta koja se uspješno izvrši s rezultatom `true`:

$$Eval(\texttt{scriptSig}||\texttt{scriptPubKey}) = \texttt{true}.$$

Cilj: malo fleksibilnosti prilikom definiranja tko i kako može potrošiti novi novčić.

Script jezik

- Jednostavan programski jezik baziran na stogu.
- Naredbe za aritmetičke i logičke operacije, manipulaciju stoga, pozive kriptografskih funkcija.
- Skripta se čita slijeva na desno, podaci se stavljaju na stog, naredbe se izvršavaju s parametrima na vrhu stoga, rezultat naredbe se opet stavlja na stog.
- Nema petlji, svaki program uvijek završava.

Tko izvršava skripte i kada?

Rudari i svi ostali čvorovi prilikom provjere ispravnosti transakcije.

Dva ishoda izvršavanja:

- Skripta se izvrši do kraja, na vrhu stoga je True (bilo što osim 0):
  - Provjera je uspješna!
- Greška pri izvršavanju ili nije True na vrhu stoga na kraju:
  - Transakcija nije ispravna.

scriptPubKey

```
<pubKey> OP_CHECKSIG
```

scriptSig

```
<sig>
```

Stog	Skripta
ε	`<sig> <pubKey> OP_CHECKSIG`
`<sig>`	`<pubKey> OP_CHECKSIG`
`<sig> <pubKey>`	`OP_CHECKSIG`
`1`	ε

Fakultet elektrotehnike i računarstva

scriptPubKey

```
OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG
```

scriptSig

```
<sig> <pubKey>
```

Stog	Skripta
ε	<sig> ...
<sig>	<pubKey> ...
<sig> <pubKey>	OP_DUP ...
<sig> <pubKey> <pubKey>	OP_HASH160 ...
<sig> <pubKey> <pubHashA>	<pubKeyHash> ...
<sig> <pubKey> <pubHashA> <pubKeyHash>	OP_EQUALVERIFY ...
<sig> <pubKey>	OP_CHECKSIG
1	ε

Problem!

Onaj koji šalje novce mora specificirati netrivijalne skripte.

Rješenje: *PayToScriptHash* transakcija.

- Primatelj je "adresa" odnosno sažetak skripte.
- Kako bi se novci potrošili potrebno je da skripta ima zadani hash i da se sama ispravno izvrši s rezultatom 1.

scriptPubKey

```
OP_HASH160 <hashOfOwnerScript> OP_EQUAL
```

scriptSig

```
<data> ...  <data> <serializedOwnerScript>
```

scriptPubKey

```
OP_HASH160 <hashOfOwnerScript> OP_EQUAL
```

scriptSig

```
<data> ...  <data> <serializedOwnerScript>
```

Čudna implementacija!

Ovo je primjer *soft fork* promjene protokola – klijenti koji ne podržavaju *PayToScriptHash* transakcije ispravno validiraju `scriptSig`.

Bitcoin skripte – primjeri *pubkey* skripta

Spali novce (*proof of burn*)!

```
OP_RETURN
```

Baci novce kroz prozor!

ε

Nagrada za koliziju u hash funkciji!

```
OP_2DUP OP_EQUAL OP_NOT OP_VERIFY OP_SHA1 OP_SWAP
OP_SHA1 OP_EQUAL
```

Možes dobiti novce tek kad si punoljetan!

```
<expiry time> OP_CHECKLOCKTIMEVERIFY OP_DROP ...
```

Problem: Tko će preuzeti rizik prevare kod kupnje?

- Ana ne želi platiti Branku prije nego što dobije robu.
- Branko ne želi poslati robu prije nego što mu je plaćeno.

Plaćanje osigurava treća strana kojoj oboje vjeruju (*escrow*):

- Transakcija je ispravna ako su je potpisali točno dvije od sljedeće tri osobe: Ana, Branko, Medijator.

scriptPubKey

```
2 <pkAna> <pkBranko> <pkMedijator> 3 OP_CHECKMULTISIG
```

scriptSig

```
OP_0 <sig1> <sig2>
```

Puno malih transakcija je neisplativo!

Ana plaća Branku 1 satoshi (10^{-8} BTC) za svaku SMS poruku.

Kumulativne transakcije:

- Na početku godine Ana pošalje 1000 satoshi-a u transakciji t.
- Potreban je potpis Ane i Branka bi se potrošio *output* od t.
- Svaku SMS poruku Ana plaća tako da Branku pošalje:
  1. "Inputs: t, Outputs: $1 \to pk_{Branko}$, $999 \to pk_{Ana}$, sig_{Ana}".
  2. "Inputs: t, Outputs: $2 \to pk_{Branko}$, $998 \to pk_{Ana}$, sig_{Ana}".
  3. "Inputs: t, Outputs: $3 \to pk_{Branko}$, $997 \to pk_{Ana}$, sig_{Ana}".
  4. ...
- Na kraju godine Branko potpiše i objavi zadnju transakciju.

Što ako Branko ne objavi niti jednu transakciju!

Ana ostaje bez 1000 satoshia.

Branko unaprijed potpiše transakciju za povrat depozita.

- Transakcija je *zaključana* – ne može se dodati u lanac prije određenog vremena.
- Parametar `locktime` definira najraniji blok ili najranije vrijeme.


```
"txid": "19d9abb1f1658472ff6ef366ae05d6b450b727e70ef0639ef43b3ed9f7f3
"hash": "19d9abb1f1658472ff6ef366ae05d6b450b727e70ef0639ef43b3ed9f7f3
"version": 2,
"size": 373,
"vsize": 373,
"locktime": 546179,
```


Više možete pronači ovdje i ovdje

Standardne transakcije

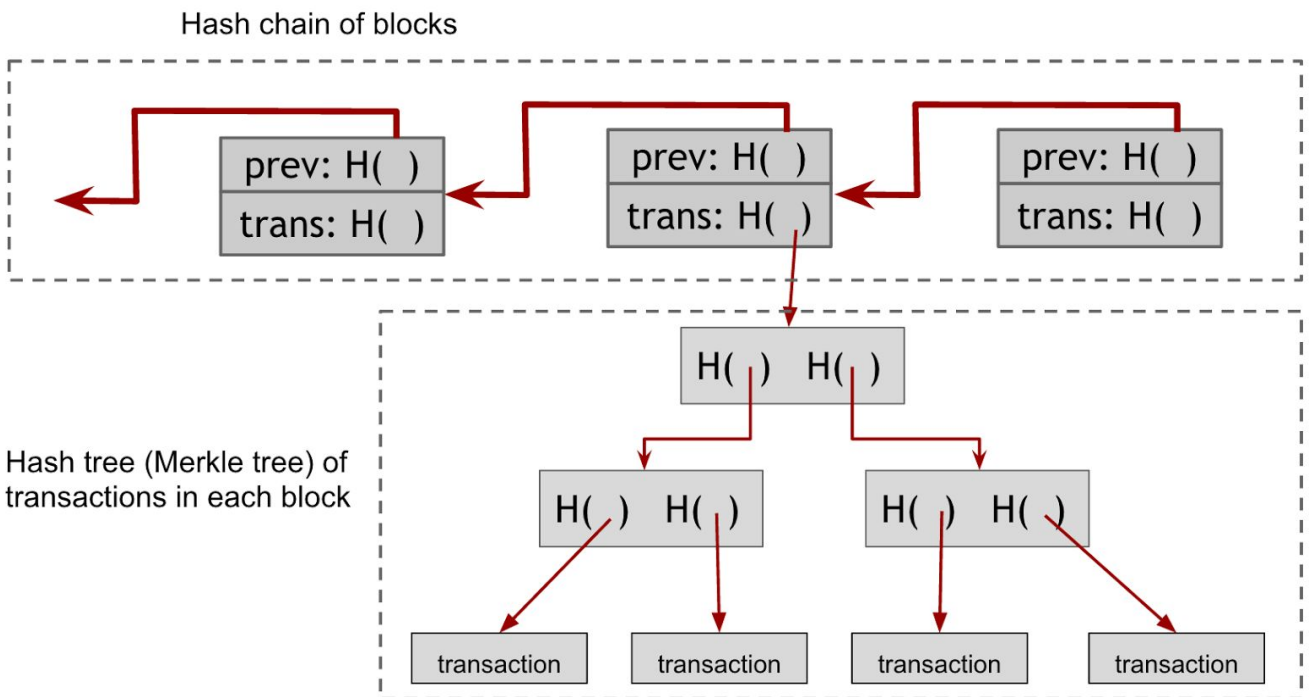
- `TX_PUBKEY`, `TX_PUBKEYHASH`, `TX_SCRIPTHASH`, `TX_MULTISIG`, `TX_NULL_DATA`
- Standardni rudari će uključivati samo standardne transakcije.
- Standardni klijenti će odašiljati samo standardne transakcije (ali će prihvatiti ispravne nestandardne transakcije u bloku).
- Proširenje (2017): Transakcije s izvojenim svjedokom (*segregated witness*) – P2WPKH, P2WSH.

Sažetak

- Bitcoin skripte omogućavaju neke zanimljive scenarije.
- U praksi je velika većina transakcija standardna.
- Pametni ugovori – motivacija za druge sustave.

Bitcoin blok

- Transakcije u Merkleovom stablu.
- Zaglavlje bloka sadrži korijen Merkleovog stabla.
- Sažetak bloka je sažetak *samo njegovog zaglavlja*.



Izvor: `bitcoinbook.cs.princeton.edu`

Bitcoin blokovi – *header*


```
"hash": "00000000000000000001dc1e0ca6db65957aee6c871f04847adb9532dab993
"ver": 536870912,
"prev_block": "00000000000000000023dfabfe863b7f9da60189d9b712abb5abfdc7
"mrkl_root": "cb39f44de7c937d5ae4d3e2fa31a49de18a39a74603790320ffc51826
"time": 1539800959,
"bits": 388350353,
"nonce": 2280495121
```

Bitcoin blokovi – *coinbase* transakcija


```
"txid": "e37cb509662912063aaf714ce074f660f3d90e9c6d66f10c9a975692210a07
"vin": [
  {
    "coinbase": "03165808162f5669614254432f4d696e65642062792077787736c2f
    "sequence": 4294967295
  }
],
"vout": [
  {
    "value": 12.50036972,
    "n": 0,
    "scriptPubKey": {
      "asm": "OP_DUP OP_HASH160 536ffa992491508dca0354e52f32a3a7a679a53
      "hex": "76a914536ffa992491508dca0354e52f32a3a7a679a53a88ac",
      "reqSigs": 1,
      "type": "pubkeyhash",
      "addresses": [
        "18cBEMRxXHqzWWCxZNtU91F5sbUNKhL5PX"
      ]
    }
  },
...
```

Novije bitne promjene...

Segreggated Witness

- BIP141, BIP143, BIP144, BIP148.
- Aktiviran 2017.

Schnorr/Taproot

- BIP340, BIP341, BIP342.
- Aktivacija u 2021.

Segreggated Witness

BIP 141 omogućuje da se potpisi (točnije `scriptSig`) izostavi iz transakcije i stavi u posebnu strukturu podataka.

Motivacija

- ECDSA ima *malleability* svojstvo — moguće je promijeniti bitove potpisa, a da on još uvijek bude ispravan. Stoga, identifikator (hash) potpisane transakcije može biti različit od identifikatora transakcije koja završi u lancu.
- Prijenos potpisa nije uvijek nužan, npr. čvor možda samo želi listu svih transakcija, bez da provjerava potpise.
- Odvojena struktura olakšava daljnji razvoj protokola. Sitne promjene u semantici skripti.

PayToWitnessPublicKeyHash (P2WPKH)

```
scriptPubKey: OP_0 <20-byte-key-hash>
scriptSig: ε
witness: <signature> <pubkey>
```

PayToWitnessScriptHash (P2WSH)

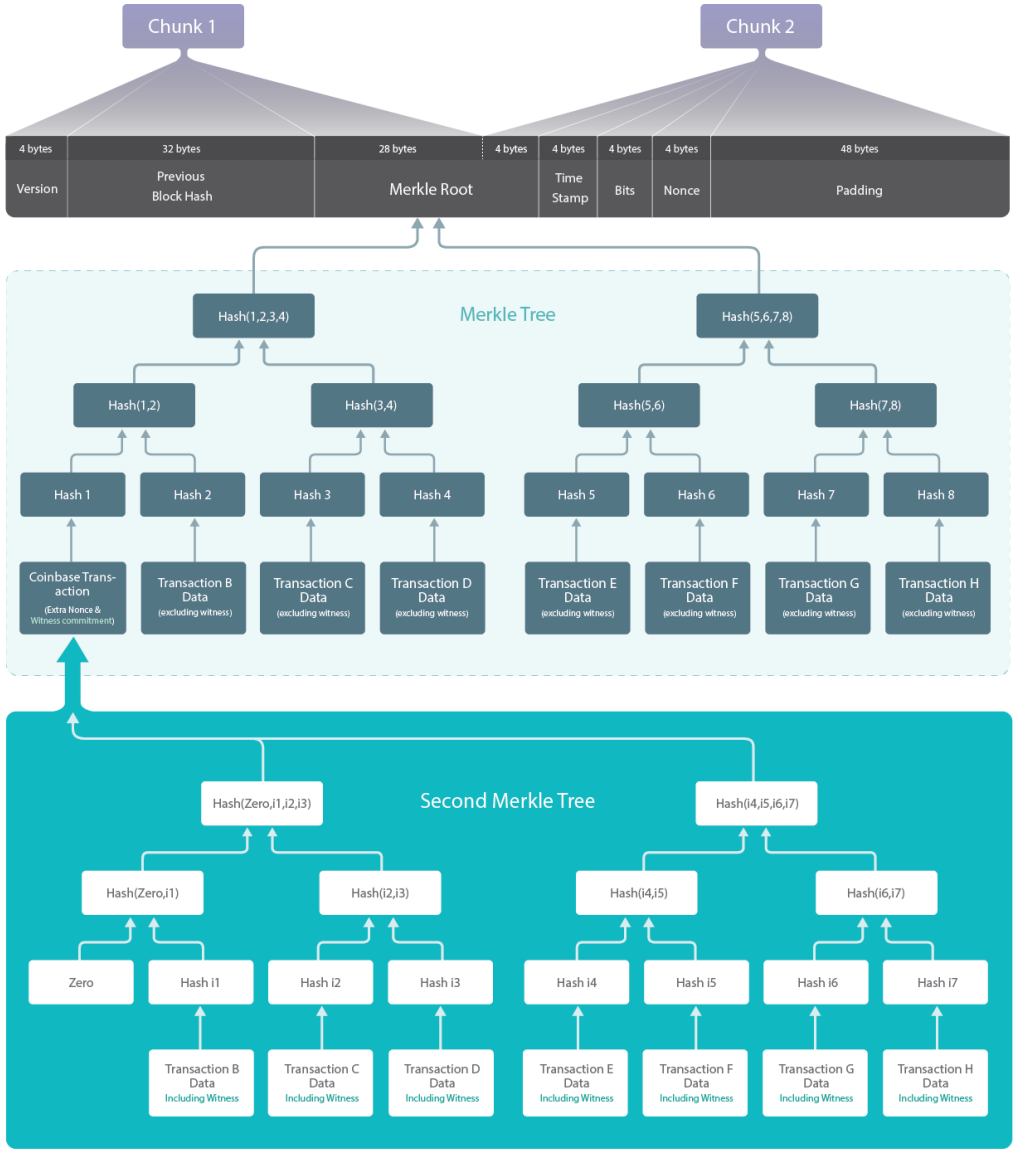
```
scriptPubKey: OP_0 <32-byte-script-hash>
scriptSig: ε
witness: <data> ...  <data> <serializedOwnerScript>
```

Dodatno, obje vrste transakcija mogu biti sadržane u *PayToScriptHash* transakciji.

Izdvojeni svjedok – stablo svjedoka

Stablo svjedoka

- Svjedoci svih transakcija su pohranjeni u Merkleovo stablo.
- Korijen stabla pohranjen u bloku, ali indirektno kako bi se radilo o *soft-fork* promjeni.



Izvor: `bitmex.com`

Schnorrovi potpis

- Sustav digitailnih potpisa sličan ECDSA sa sličnim sigurnosnim svojstvima.
- Za razliku od ECDSA, Schnorrovi potpisu su *non-malleable*.
- Omogućava *agregaciju* ključeva i potpisa kao alternativu *MultiSig* transakcijama.

Agregacija

- Ana ima par ključeva (sk_A, pk_A)
- Branko ima par ključeva (sk_B, pk_B)
- Javni ključevi se mogu agregirati u javni ključ pk_{AB}.
- Postoji protokol kojim Ana i Branko mogu zajednički potpisati poruku m i dobiti potpis σ_{AB}.
- Verifikacija potpisa je jednaka kao i da nema agregacije.

Napredne upotrebe

- Moguće je agregaciju poopćiti u k-od-n sheme.
- *Atomic swaps*, *Payment channels*, ...

Prednosti

- Efikasnost – jednostavnije je provjeriti potpis, nema potrebe za duljim MULTISIG skriptama.
- Privatnst – javni ključ vlasnika novčića izgleda kao *obični* javni ključ.

Merkle Abstract Syntax Tree

- Skripta za trošenje opisana Merkleovim stablom.
- *Vlasnik* novčića je hash pokazivač na korijen stabla.
- Prilikom trošenja kao dokaz se prilažu samo potrebni fragmenti stabla.

S_1: Charlie može potrošiti nakon 2030.
S_2: Ana i Branko mogu zajedno potrošiti nakon 2025.
S_3: Dvoje od {Ana, Branko, Charlie} mogu potrošiti nakon 2022.
S_4: Ana, Branko i Charlie mogu zajedno potrošiti novčić bilo kad.

Primjer mogućeg korištenja u kriptovaluti

`M`: MAST kojem su S_1, S_2, S_3, S_4, listovi.
`scriptPubKey: <hash-pokazivac-na-vrh-stabla-M>`
`witness: <put-u-M-do-S_i> <S_i> <ulaz-za-S_i>`

Taproot

- Postoji način da se matematički kombinira (*tweak*) javni ključ i vrh Merkleovog stabla.
- Vlasnik novčića se specificira samo javnim ključem, a prilikom trošenja se može ispostaviti da se zapravo koristi skripta koja je dio MAST-a, s čijim korijenom je agregiran javni ključ.

Taproot (pojednostavljeno)

`M`: MAST kojem su S_1, S_2, S_3, S_4, listovi.
pk^M_{ABC}: Javni ključ tweak(agg(pk_A, pk_B, pk_C), M).
`scriptPubKey`: pk^M_{ABC}
`witness`: `<signature>` (kada troše svi zajedno)
`witness`: `<witness-za-M> <dokaz-da-je-kljuc-kombiniran-s-M>`

Sažetak

- Svaki novčić se može potrošiti ili potpisom (koristeći Schnorrove potpise) ili skriptom u MAST-u.
- Svi novčići (odnosno outputi koji koriste Taproot shemu) izgledaju *jednako* – sadrže jedan javni ključ, za kojeg se može ispostaviti da zapravo "skriva" korijen MAST stabla.
- Analizom se ne može utvrditi da li (i koje) skripte kontroliraju trošenje.

Prednosti

- Veća privatnost.
- Veća efikasnost.

Izvor: `taproot.watch`